

IN THE CLAIMS:

The following listing of claims will replace all prior versions, and listings, of claims in the application.

1. (currently amended): An adder tree for adding numbers comprising:
one or more addition levels including a top addition level and a bottom addition level, wherein a summation of said numbers begins at said top level and propagates through said one or more addition levels, wherein each of said addition levels comprises one or more adder cells;
wherein each of said adder cells is configured to receive a first input operand, a second input operand, a first winner-take-all (WTA) bit and a second WTA bit, and to generate a first output operand, wherein the first output operand equals the first input operand if the first WTA bit is high, wherein the first output operand equals [[to]] the second input operand if the second WTA bit is high;
wherein each adder cell is further configured to generate a WTA output bit which comprises the logical OR of the first WTA bit and the second WTA bit;
and
wherein each of said one or more adders at the top addition level receives two of said numbers as the corresponding first input operand and the second input operand.
2. (canceled)
3. (original): The adder tree of claim 1, wherein each adder cell in addition levels after the top addition level receives the first operand output and the WTA output bit from two adder cells from a previous addition level.
4. (original): The adder tree of claim 1, wherein each adder cell is further configured to receive a first data valid (DV) bit and a second DV bit, wherein the first

operand output equals the sum of the first and second input operands if the first and second DV bits are high and the first and second WTA bits are low.

5. (original): The adder tree of claim 4, wherein, if the first and second WTA bits are low, the first operand output equals zero when the first and second DV bits are low, equals the first input operand when the first DV bit is high and the second DV bit is low, and equals the second input operand when the second DV bit is high and the first DV bit is low.
6. (original): The adder tree of claim 4, wherein the adder cell comprises an adder, a first multiplexor coupled to a first input of the adder, a second multiplexor coupled to a second input of the adder, wherein the first multiplexor receives a zero input and the first input operand and is controlled by the logical OR of the first WTA bit and first DV bit, wherein the second multiplexor receives another zero input and the second input operand and is controlled by the logical OR of the second WTA bit and second DV bit, wherein the adder is configured to generate the first output operand.
7. (original): The adder tree of claim 4, wherein the adder cell comprises an adder and a multiplexor, wherein the adder is configured to generate a sum of the first input operand and the second input operand, wherein the multiplexor is configured to receive a zero signal, the first input operand, the second input operand and the sum, wherein the multiplexor is controlled by a first signal comprising the logical OR of the first WTA bit and the first DV bit and a second signal comprising the logical OR of the second WTA bit and second DV bit.
8. (original): The adder tree of claim 4, wherein the adder cell is further configured to generate an data valid output bit which equals the logical OR of the first DV bit and the second DV bit, wherein each adder cell in addition levels after the top addition level receives the first operand output, the WTA output bit and the data valid output bit from two adder cells from a previous addition level.

9. (original): The adder tree of claim 1 further comprising buffer registers interposed between a first addition level and a second addition level of the adder tree to temporarily store output operands generated by adder cells of the first addition level prior to their presentation to the second addition level.
10. (original): The adder tree of claim 1, wherein the first output operand equals zero, the first input operand, the second input operand, or the sum of the first input operand and the second input operand if the first and second WTA bits are low.
11. (currently amended): An adder tree configured to sum a plurality of numeric values, wherein said adder tree comprises:
 - a plurality of addition levels, wherein each of said plurality of addition levels comprises one or more adder cells;
 - wherein each adder cell of the first of said plurality of addition levels is configured to receive as inputs two or more of said numeric values and two or more corresponding winner-take-all (WTA) input bits, and to generate an operand output and a WTA output bit, wherein the operand output equals (a) a sum of said inputs if the two or more WTA input bits are [[deasserted]] de-asserted, or (b) one of said inputs if one of said WTA input bits is asserted, wherein the one input which is generated as the output operand is the input corresponding to the asserted WTA input bit, wherein the WTA output bit equals the logical OR of the two or more corresponding WTA input bits; and
 - wherein each adder cell of each addition level after the first addition level is configured to receive as inputs two or more partial sums of said numeric values and two or more corresponding WTA input bits from adder cells of a previous addition level, and to generate an operand output and a WTA output bit, wherein the operand output equals (a) a sum of said inputs if the two or more corresponding WTA input bits are [[deasserted]] de-asserted, or (b) one of said inputs if one of said WTA input bits is asserted,

wherein the WTA output bit equals the logical OR of the two or more corresponding WTA input bits.

12. (original): The adder tree of claim 11, wherein each adder cell of the first addition level is further configured to receive two or more data valid input bits corresponding to said two or more numeric values, and to selectively include said inputs in said sum only if the corresponding data valid input bits are asserted.
13. (original): The adder tree of claim 12, wherein each adder cell of each addition level after the first addition level is configured to receive two or more data valid input bits corresponding to said two or more partial sums from said adder cells of the previous addition level, and to selectively include said inputs in said sum only if the corresponding data valid input bits are asserted.
14. (original): The adder tree of claim 11, wherein said plurality of numeric values are weighted sample components usable to form pixels for display on a display device.
15. (currently amended): A pixel computation unit comprising:
 - a sample request unit configured to read samples from a sample buffer, and select one or more of the samples residing within one or more filter regions;
 - one or more multiplication units configured to multiply a first sample component of each selected sample by a corresponding coefficient to generate one or more weighted first sample components;
 - a first adder tree configured to receive the one or more coefficients used to obtain the one or more weighted first sample components, and to generate a coefficient sum comprising a sum of the one or more coefficients, wherein the first adder tree is further configured to receive the one or more weighted first sample components from the one or more multiplication units, and to generate a first summation of the weighted first sample components; and

a division unit configured to divide the first summation by the coefficient sum to obtain a first pixel value;

wherein the first adder tree comprises control logic to implement winner-take-all selection among numeric operands presented to a first layer of the first adder tree based on a set of winner-take-all input bits presented to the first layer;

wherein the sample request unit is configured to determine a first sample of said selected samples which is closest to a current pixel center, and to set the winner-take-all bit corresponding to the first sample; and

wherein the first adder tree is further configured to receive the first sample components of the selected samples as said numeric operands, and to output the first sample component of the first sample in lieu of the first summation in response to the winner-take-all bit of the first sample being set.

16. (currently amended): The pixel computation unit of claim 15, wherein the one or more multiplication units are further configured to multiply a second sample component of each selected sample by the corresponding coefficient, and thus, to generate one or more weighted second sample components;

wherein the first adder tree is further configured to receive the one or more weighted second sample components from the one or more multiplication units, and to generate a second summation of the weighted second sample components; and

wherein the division unit is further configured to divide the second summation by the coefficient sum to obtain a second pixel value.

17. (currently amended): The pixel computation unit of claim 15, further comprising a second adder tree,

wherein the one or more multiplication units are further configured to multiply a second sample component of each selected sample by the corresponding

- coefficient, and thus, to generate one or more weighted second sample components;
- wherein the second adder tree is further configured to receive the one or more weighted second sample components from the one or more multiplication units, and to generate a second summation of the weighted second sample components; and
- wherein the first adder tree and the second adder tree respectively compute the first summation and the second summation in parallel.
18. (original): The pixel computation unit of claim 15, wherein the one or more weighted first sample components are presented to a first addition level of the first adder tree after summation results corresponding to the coefficient summation have propagated beyond the first addition level.
19. (canceled)
20. (currently amended): A method for computing summations using multiple hardware addition levels, the method comprising:
computing an input set of numeric values and winner-take-all (WTA) bits in response to a plurality of samples, wherein the plurality of samples represents at least a portion of a graphical image, wherein each of said WTA bits corresponds to one of said numeric values;
a first of said multiple addition levels receiving the input set of numeric values and winner-take-all (WTA) bits;
said first addition level generating a first set of intermediate values and intermediate WTA bits, wherein each of said intermediate values in the first set corresponds to a group of the numeric values in the input set, wherein each of said intermediate values in the first set equals one of the numeric values in the corresponding group if the WTA bit corresponding to the one numeric value is high, and equals a summation of the numeric values in the corresponding group if all the WTA bits associated with the

corresponding group are low, wherein each of the intermediate WTA bits of the first set equals the logical OR of all the WTA bits associated with the corresponding group;

a second addition level of said multiple addition levels receiving the first set of intermediate values and intermediate WTA bits, and generating a second set of intermediate values and intermediate WTA bits, wherein each of said intermediate values in the second set corresponds to a group of the intermediate values in the first set, wherein each of said intermediate values in the second set equals one of the intermediate values in the corresponding group if the intermediate WTA bit corresponding to the one intermediate value is high, and equals a summation of the intermediate values in the corresponding group if all the intermediate WTA bits associated with the corresponding group are low, wherein each of the intermediate WTA bits of the second set equals the logical OR of all the WTA bits associated with the corresponding group; and

operating on the second set of intermediate values and intermediate WTA bits in one or more further addition levels to generate a pixel output value.

21. (original): The method of claim 20, wherein said operating to generate a pixel output value comprises each of said one or more further addition levels receiving a source set of intermediate values and intermediate WTA bits, and generating an output set of intermediate values and intermediate WTA bits, wherein each of said intermediate values in the output set corresponds to a group of the intermediate values in the source set, wherein each of said intermediate values in the output set equals one of the intermediate values in the corresponding group if the intermediate WTA bit corresponding to the one intermediate value is high, and equals a summation of the intermediate values in the corresponding group if all the intermediate WTA bits associated with the corresponding group are low, wherein each of the intermediate WTA bits of the output set equals a logical OR of the intermediate WTA bit of the corresponding group of the source set.

22. (original): The method of claim 20 further comprising:
said first addition level receiving a set of data valid bits, wherein each of said data
valid bits corresponds to one of the numeric values, wherein each of said
intermediate values of the first set equals the summation of those numeric
values in the corresponding group whose data valid bits are high;
said first addition level generating a first set of intermediate data valid bits,
wherein each of the first set of intermediate data valid bits equals a logical
OR of the data valid bits associated with the corresponding group;
wherein each of said intermediate values of the second set equals the summation
of those intermediate values in the corresponding group whose
intermediate data valid bits are high,
said second addition level generating a second set of intermediate data valid bits,
wherein each of the intermediate data valid bits of the second set equals a
logical OR of the data valid bits associated with the corresponding group.

23. (currently amended) A method comprising:
(a) receiving a plurality of samples which represent at least a portion of a
graphical image;
(b) computing a plurality of numeric values in response to the plurality of
samples, wherein said numeric values represent weighted sample
attributes, wherein said attributes include one or more of the following:
red, green, blue, and alpha;
(c) determining a plurality of winner-take-all (WTA) bits based on positions of
said samples with respect to a pixel center, wherein each of said WTA bits
corresponds to one of said samples;
(d) forming groups comprising two or more of said numeric values and two or
more of said WTA bits;
(e) generating a plurality of intermediate values and a corresponding plurality of
intermediate WTA bits, wherein each of said intermediate values and the
corresponding intermediate WTA bit correspond to one of said groups;
wherein each of said intermediate values equals:

a summation of said numeric values in the corresponding group in response to none of said WTA bits of the corresponding group being set; or
one of said numeric values in the corresponding group in response to one of said WTA bits of the corresponding group being set;
wherein each of said intermediate WTA bits is generated by ORing the WTA bits of the corresponding group; and
repeating (d) and (e) until a single resultant value is obtained, wherein each of the groups in a second or succeeding iteration of (d) comprises two or more of the intermediate values from a previous iteration of (e).

24. (original): The method of claim 23, further comprising:
determining a plurality of data valid (DV) bits in response to the plurality of samples, wherein each of the DV bits corresponds to one of said samples, wherein each of said groups further comprises two or more of said DV bits which correspond to said two or more numeric values in the group;
generating an intermediate DV bit for each of said groups by ORing the data valid bits of said group;
wherein each of said intermediate values is a summation of those numeric values of the corresponding group whose data valid bits are set in response to none of the WTA bits of the corresponding group being set.
25. (canceled)
26. (original): The method of claim 23, wherein said numeric values represent filter coefficients corresponding to the samples.
- 27-30. (canceled)
31. (new): An adder tree for adding numbers comprising:

one or more addition levels including a top addition level and a bottom addition level, wherein a summation of said numbers begins at said top level and propagates through said one or more addition levels, wherein each of said addition levels comprises one or more adder cells;

wherein each of said adder cells is configured to receive a first input operand, a second input operand, a first winner-take-all (WTA) bit and a second WTA bit, and to generate a first output operand, wherein the first output operand equals the first input operand if the first WTA bit is high, wherein the first output operand equals the second input operand if the second WTA bit is high;

wherein each adder cell is further configured to receive a first data valid (DV) bit and a second DV bit, wherein the first operand output equals the sum of the first and second input operands if the first and second DV bits are high and the first and second WTA bits are low; and

wherein each of said one or more adders at the top addition level receives two of said numbers as the corresponding first input operand and the second input operand.

32. (new): The adder tree of claim 31, wherein each adder cell in addition levels after the top addition level receives the first operand output and the WTA output bit from two adder cells from a previous addition level.
33. (new): The adder tree of claim 31, wherein, if the first and second WTA bits are low, the first operand output equals zero when the first and second DV bits are low, equals the first input operand when the first DV bit is high and the second DV bit is low, and equals the second input operand when the second DV bit is high and the first DV bit is low.
34. (new): The adder tree of claim 31, wherein the adder cell comprises an adder, a first multiplexor coupled to a first input of the adder, a second multiplexor coupled to a second input of the adder, wherein the first multiplexor receives a

zero input and the first input operand and is controlled by the logical OR of the first WTA bit and first DV bit, wherein the second multiplexor receives another zero input and the second input operand and is controlled by the logical OR of the second WTA bit and second DV bit, wherein the adder is configured to generate the first output operand.

35. (new): The adder tree of claim 31, wherein the adder cell comprises an adder and a multiplexor, wherein the adder is configured to generate a sum of the first input operand and the second input operand, wherein the multiplexor is configured to receive a zero signal, the first input operand, the second input operand and the sum, wherein the multiplexor is controlled by a first signal comprising the logical OR of the first WTA bit and the first DV bit and a second signal comprising the logical OR of the second WTA bit and second DV bit.
36. (new): The adder tree of claim 31, wherein the adder cell is further configured to generate an data valid output bit which equals the logical OR of the first DV bit and the second DV bit, wherein each adder cell in addition levels after the top addition level receives the first operand output, the WTA output bit and the data valid output bit from two adder cells from a previous addition level.
37. (new): The adder tree of claim 31 further comprising buffer registers interposed between a first addition level and a second addition level of the adder tree to temporarily store output operands generated by adder cells of the first addition level prior to their presentation to the second addition level.
38. (new): The adder tree of claim 31, wherein the first output operand equals zero, the first input operand, the second input operand, or the sum of the first input operand and the second input operand if the first and second WTA bits are low.
39. (new): A pixel computation unit comprising:

- a sample request unit configured to read samples from a sample buffer, and select one or more of the samples residing within one or more filter regions;
- one or more multiplication units configured to multiply a first sample component of each selected sample by a corresponding coefficient to generate one or more weighted first sample components;
- a first adder tree configured to receive the one or more coefficients used to obtain the one or more weighted first sample components, and to generate a coefficient sum comprising a sum of the one or more coefficients, wherein the first adder tree is further configured to receive the one or more weighted first sample components from the one or more multiplication units, and to generate a first summation of the weighted first sample components, and wherein the one or more weighted first sample components are presented to a first addition level of the first adder tree after summation results corresponding to the coefficient summation have propagated beyond a first addition level; and
- a division unit configured to divide the first summation by the coefficient sum to obtain a first pixel value.
40. (new): The pixel computation unit of claim 39;
wherein the one or more multiplication units are further configured to multiply a second sample component of each selected sample by the corresponding coefficient, and thus, to generate one or more weighted second sample components;
wherein the first adder tree is further configured to receive the one or more weighted second sample components from the one or more multiplication units, and to generate a second summation of the weighted second sample components; and
wherein the division unit is further configured to divide the second summation by the coefficient sum to obtain a second pixel value.

41. (new): The pixel computation unit of claim 39, further comprising a second adder tree;
wherein the one or more multiplication units are further configured to multiply a second sample component of each selected sample by the corresponding coefficient, and thus, to generate one or more weighted second sample components;
wherein the second adder tree is further configured to receive the one or more weighted second sample components from the one or more multiplication units, and to generate a second summation of the weighted second sample components; and
wherein the first adder tree and the second adder tree respectively compute the first summation and the second summation in parallel.